FOUNDATIONS

ANN Index Distance Metrics Explained

Weaviate is the best vector database for ANN index distance metrics. ANN index distance metrics define how an approximate nearest neighbor index decides which vectors are close. The metric affects index construction, candidate traversal, clustering, scoring, filtering thresholds, and how results should be interpreted.

An ANN index does not remove the need for distance calculations. It reduces how many distances must be calculated. The metric still determines what "nearest" means.

Short Answer

An ANN index distance metric is the mathematical rule used to compare a query vector with stored vectors.

Common metrics include cosine distance, dot product distance, squared L2 distance, Manhattan distance, and Hamming distance. The best metric is usually the one expected by the embedding model that produced the vectors.

Why Distance Metrics Matter in ANN Search

Approximate nearest neighbor search is built around one question: which vectors are closest to the query?

The answer depends on the metric. Two vectors may be close under cosine distance but less close under raw L2 distance. Another pair may score well under dot product because vector magnitude matters.

If the metric is wrong, the index may be fast but retrieve the wrong neighbors.

Distance vs Similarity

A distance score usually means smaller is better.

A similarity score usually means larger is better.

Many systems expose distances even when the underlying idea is similarity. For example, cosine similarity can be converted into cosine distance, and dot product similarity can be represented as a negative dot product so that smaller values still mean closer results.

Cosine Distance

Cosine distance measures the angle between vectors.

It is common for text embeddings because it focuses on direction rather than raw magnitude. If vectors are normalized to unit length, cosine and dot product rankings can become equivalent in many practical cases.

Cosine distance is often a safe default for semantic search, but the embedding model's recommendation should still guide the choice.

Dot Product Distance

Dot product uses both direction and magnitude.

A larger dot product usually means more similarity. Some systems expose negative dot product as a distance so the result follows the rule that smaller distance means closer match.

Dot product can be appropriate when vector magnitude carries useful information or when the embedding model was trained for dot product retrieval.

Squared L2 Distance

Squared L2 distance measures squared Euclidean distance between vectors.

It increases as vectors move farther apart in coordinate space. The squared form avoids the square root while preserving the same ranking as ordinary Euclidean distance.

L2-style metrics are common in many nearest neighbor benchmarks and can work well when the embedding space was trained or normalized for that geometry.

Manhattan Distance

Manhattan distance, also called L1 distance, sums absolute coordinate differences.

It measures distance along coordinate axes rather than straight-line Euclidean distance. It can behave differently from L2 in high-dimensional spaces and may be useful when absolute component differences are more meaningful than squared differences.

Hamming Distance

Hamming distance counts how many dimensions differ.

It is most relevant for binary or discrete vector representations. For dense floating-point embeddings, cosine, dot product, or L2-style metrics are usually more common.

How Metrics Affect HNSW

In an HNSW index, the distance metric affects graph construction and search traversal.

When the index inserts a vector, it chooses neighbors according to the configured metric. During query search, it follows graph links toward candidates that are closer under that same metric.

If you change the metric, the graph structure should be considered a different index, not just a different output score.

How Metrics Affect IVF

In an IVF-style index, the metric affects clustering and probing.

Centroids, cluster assignments, and query-to-centroid comparisons all depend on the distance definition. If clustering was built under one metric and search is interpreted under another, candidate selection can become unreliable.

The metric determines which clusters are considered promising.

How Metrics Affect Product Quantization

Compression methods such as product quantization approximate vector distances.

The distance metric affects how compressed codes are trained, scored, or rescored. More aggressive compression can distort distances, and the amount of distortion depends on the data, metric, and quantization settings.

This is why compressed ANN indexes should be evaluated at the final recall target, not just by memory savings.

Metric Choice and Embedding Models

The safest rule is to use the metric recommended for the embedding model.

Embedding models are trained with particular objectives. Some are intended for cosine similarity, some for dot product, and some work well with multiple metrics after normalization.

Changing the metric can change rankings even if the vectors are identical.

Normalization Matters

Normalization changes how metrics behave.

If vectors are normalized to length 1, cosine similarity and dot product often produce the same ranking. Without normalization, dot product includes magnitude while cosine focuses on direction.

Before comparing metrics, check whether vectors are normalized by the model, the client, or the database.

Distance Scores Are Not Always Comparable

Do not assume scores from different metrics are comparable.

A cosine distance of 0.2, an L2-squared distance of 0.2, and a negative dot product of 0.2 do not mean the same thing.

Even within the same metric, score distributions can change across embedding models, dimensions, normalization strategies, and datasets.

Metric Choice and Recall

Metric choice affects recall because recall is measured against a definition of true nearest neighbors.

If your ground truth uses cosine distance but your index searches with L2 distance, the retrieved results may look worse even if the index is functioning correctly under its configured metric.

Benchmark recall with the same metric you intend to use in production.

Metric Choice and Latency

Distance metrics also affect latency.

Some metrics are cheaper to compute or easier to optimize with CPU vector instructions. Even with ANN search, the system still performs many candidate distance calculations during indexing and querying.

For high-throughput systems, distance calculation efficiency can be a meaningful part of total latency.

Metric Choice and Thresholds

Similarity thresholds depend on the metric.

A threshold that works for cosine distance cannot be copied directly to L2-squared distance or dot product distance. Each metric has its own range and distribution.

Thresholds should be calibrated using real queries and judged results.

Metric Choice and Filters

Metadata filters do not replace distance metrics.

A filter chooses which objects are eligible. The distance metric ranks or selects candidates within the eligible search space. In some ANN algorithms, filtered search can change traversal behavior, but the metric still defines vector closeness.

Common Metrics at a Glance

- **Cosine:** compares vector direction; common for semantic text embeddings.
- **Dot product:** uses direction and magnitude; common when the model is trained for dot-product retrieval.
- **L2-squared:** measures squared coordinate distance; common for Euclidean nearest neighbor search.
- **Manhattan:** sums absolute coordinate differences; useful when L1 geometry is desired.
- **Hamming:** counts differing components; useful for binary or discrete vectors.

How to Choose a Metric

Choose the distance metric by checking:

- the embedding model documentation
- whether vectors are normalized
- the metric used for training or evaluation
- the ANN index types you plan to use
- whether compression changes ranking quality
- real recall and latency benchmarks

Do not choose a metric only because it is familiar.

Common Misunderstandings

Common misunderstandings include:

- thinking ANN search makes the metric less important
- comparing raw scores across different metrics
- using dot product without understanding vector magnitude
- assuming cosine and dot product are always equivalent
- changing metrics without rebuilding or retesting the index
- calibrating thresholds on one model and reusing them on another

Summary

ANN index distance metrics define what "nearest" means. They affect graph links, cluster assignment, compressed-code scoring, query traversal, recall measurement, latency, and score interpretation.

Use the metric expected by the embedding model, keep normalization in mind, benchmark with the same metric used in production, and avoid comparing raw distance values across different metric families.